-NoValue-

No Title Given

No Author Given

Abstract T

he Bioconductor project enters its third decade with over two thousand packages for genomic data science, over 100,000 annotation and experiment resources, and a global system for convenient distribution to researchers. Over 60,000 PubMed Central citations and terabytes of content shipped per month attest to the impact of the project on cancer genomic data science. This report provides an overview of cancer genomics resources in Bioconductor. After an overview of Bioconductor project principles, we address exploration of institutionally curated cancer genomics data such as TCGA. We then review genomic annotation and ontology resources relevant to cancer and then briefly survey Analytical workflows addressing specific topics in cancer genomics. Concluding sections cover how new software and data resources are brought into the ecosystem and how the project is tackling needs for training of the research workforce. Bioconductor's strategies for supporting methods developers and researchers in cancer genomics are evolving along with experimental and computational technologies. All the tools described in this report are backed by regularly maintained learning resources that can be used locally or in cloud computing environments.

introduction

1 Introduction

Computation is a central component of cancer genomics research. Tumor sequencing is the basis of computational investigation of mutational, epigenetic and immunologic processes associated with cancer initiation and progression. Numerous computational workflows have been produced to profile tumor cell transcriptomes and proteomes. New technologies promise to unite sequence-based characterizations with digital histopathology, ultimately driving efforts in molecule design and evaluation to produce patient-centered treatments.

Bioconductor is an open source software project with a 20 year history of uniting biostatisticians, bioinformaticians, and genome researchers in the creation of an ecosystem of data, annotation, and analysis resources for research in genome-scale biology. This paper will review current approaches of the project to advancing cancer genomics. After a brief discussion of basic principles of the Bioconductor project, we will present a "top down" survey of resources useful for cancer bioinformatics. Primary sections address

- how to explore institutionally curated cancer genomics data
- genomic annotation resources relevant to cancer genomics
- analytical workflows
- components for introducing new data or analyses
- pedagogics and workforce development.

Appendix 1 (section **??**) of this paper includes descriptions of 69 Bioconductor software packages that use the term "cancer" in their package metadata.

Appendix 2 (section **??**) of this paper includes descriptions of 63 Bioconductor experimental data packages that use the term "cancer" in their package metadata.

bioconductor-principles

2 Bioconductor principles

r-packages-and-vignettes

2.1 R packages and vignettes

Software tools and data resources in Bioconductor are organized into "R packages". These are collections of folders with data, code (principally R functions), and documentation following a protocol specified in https://cran.r-project.org/doc/manuals/R-exts.htmlWriting R Extensions. R packages have a DESCRIPTION file with metadata about package contents and provenance. Package structure can be checked for validity using the `R CMD check` facility. Documentation of code and data can be programmatically checked for existence and validity. The DESCRIPTION file for a package specifies its version and also gives precise definition of how an R package may depend upon versions of other packages.

At its inception, Bioconductor introduced a new approach to holistic package documentation called "vignette". Vignettes provide narrative and explanation interleaved with executable code describing package operations. While R function manual pages describe the operation of individual functions, vignettes illustrate the interoperation of package components and provide motivation for both package design but also context for its use.

r-package-repositories-repository-evolution

2.2 R package repositories; repository evolution

Bioconductor software forms a coherent ecosystem that can be checked for consistency of versions of all packages available in a given installation of R. Bioconductor packages may specify dependency on other Bioconductor packages, or packages that are available in the CRAN repository. Bioconductor does not include packages with dependencies on "github-only" packages. Later in this paper we will provide details on package quality assurance that provide a rationale for this restriction.

Major updates to the R language occur annually, and updates are preceded by careful assessment of effects of language change on Bioconductor package operations. These effects can be identified through changes in the output of R CMD check. The Bioconductor ecosystem is updated twice a year, once to coincide with update to R, and once about six months later. The semianual updates reflect the need to track developments in the fast-moving field of genomic data science.

package-quality-assessment-installation-consistency

2.3 Package quality assessment; installation consistency

The BiocCheck function is used to provide more stringent assessment of package compliance with basic principles of the Bioconductor ecosystem.

The BiocManager package provides for installing and updating package and has functionality for verifying the coherence and version status of the currently installed package collection. This is important in the context of a language and package ecosystem that changes every six months, while analyses may take years to complete. Tools for recreating past package collections are available to assist in reproducing outputs of prior analyses.

unifying-assay-and-sample-data-summarizedexperiment-and-multiassayexperiment

2.4 Unifying assay and sample data: SummarizedExperiment and MultiAssayExperiment

Most of the data from genome-scale experiments to be discussed in this chapter are organized in special data containers rooted in the concepts of the SummarizedExperiment class. Briefly, assay data are thought of as occupying a $G \times N$ array, and sample level data occupy an $N \times K$ table. The array and the table are linked together in the SummarizedExperiment; see Figure 1.

Multiple representations of assay results may be managed in this structure, but all assay arrays must have dimensions $G \times N$.

For experiment collections in which the same samples are subjected to multiple genome-scale assays, MultiAssayExperiment containers are used. See Figure 2 for the layout.

Further details on these data structures will be provided in section **??**.

cache

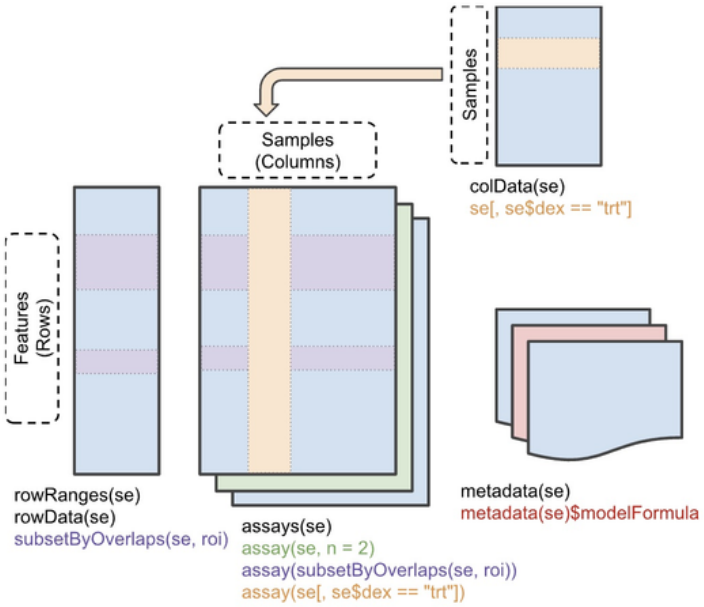

Fig. 1 SummarizedExperiment schematic.

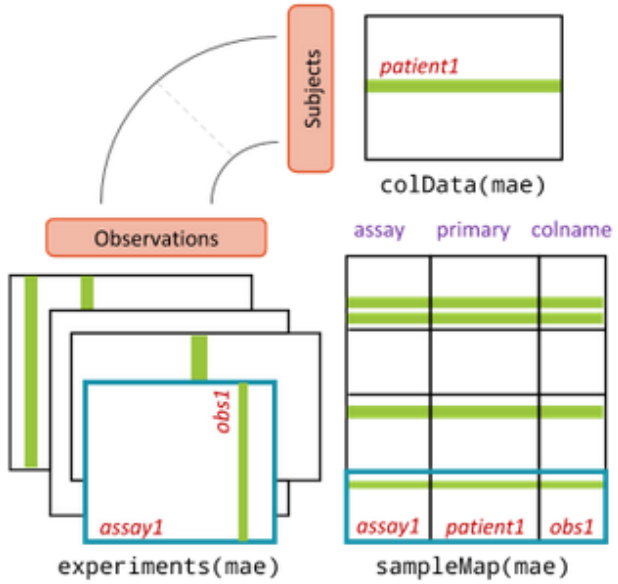

Fig. 2 MultiAssayExperiment schematic.

2.5 Downloading and caching cancer genomics data and annotations

Downloading and managing data from various online resources can be excessively time consuming. Bioconductor encourages data caching for increased efficiency and reproducibility. The caching data methods employed in Bioconductor allow analysis code to concisely refer to data resources as needed, with minimal attention to how data are stored, retrieved or transformed. It allows for easy management and reuse of data that are on remote servers or in cloud, storing source location and providing information for data updates. The BiocFileCache Bioconductor package handles data management from within R.

BiocFileCache is a general-use caching system but Bioconductor also provides "Hubs", AnnotationHub and ExperimentHub, to help distributed annotation or experimental data hosted externally. Both AnnotationHub and ExperimentHub use BiocFileCache to handle download and caching of data.

AnnotationHub provides a centralized repository of diverse genomic annotations, facilitating easy access and integration into analyses. Researchers can seamlessly retrieve information such as genomic features, functional annotations, and variant data, streamlining the annotation process for their analyses.

ExperimentHub extends this concept to experimental data. It serves as a centralized hub for storing and sharing curated experiment-level datasets, allowing researchers to access a wide range of experimental designs and conditions. This cloud-based infrastructure enhances collaboration and promotes the reproducibility of analyses across different laboratories.

The curatedTCGAData package provides some resources through ExperimentHub, as do many other self-identified "CancerData" resources. Once the ExperimentHub is loaded, it can be queried for terms of interest.

{r useeh} <!-- , fig.cap="ExperimentHub attachment, retrieval, query, and response when seeking cancer-related data.", message=FALSE} --> library(ExperimentHub) eh <- ExperimentHub() query(eh, "CancerData")

Multiple terms can be used to narrow results before choosing a download.

[] query(eh, c("cancerData", "esophageal")) ## ExperimentHub with 2 records ## # snapshotDate(): 2023-10-24 ### $dataprovider$: $UniversityofCaliforniaSanFrancisco$###species: Homo sapiens ### $rdataclass$: $RangedSummarizedExperiment, data.frame$###$additionalmcols()$: $taxonomyid$

Similarly AnnotationHub files can be downloaded for annotating data. For example, the ensembl 110 release of gene and protein annotations are obtained with the following:

[] library(AnnotationHub) ah <- AnnotationHub() tag = names(query(ah, c("Ensembl","110", "Homo sapiens"))) ens110 <- ah[[tag]]

exploring-institutionally-curated-cancer-genomics-data

3 Exploring institutionally curated cancer genomics data

the-cancer-genome-atlas